

# Segmenting Bayesian Networks for Intelligent Information Dissemination in Collaborative, Context-Aware Environments with Bayeslets

Korbinian Frank<sup>a</sup>, Matthias Röckl<sup>b</sup>, Tom Pfeifer<sup>c</sup>, Patrick Robertson<sup>a</sup>

<sup>a</sup>*Institute of Communications and Navigation*

*German Aerospace Center (DLR)*

*Oberpfaffenhofen, Germany*

<sup>b</sup>*In2Soft GmbH*

*Munich, Germany*

<sup>c</sup>*Technical University of Berlin*

*Berlin, Germany*

---

## Abstract

With ever smaller processors and ubiquitous internet connectivity, the pervasive computing environments from Mark Weiser's vision are coming closer. For their context-awareness, they will have to incorporate data from the abundance of sensors integrated in everyday life and to benefit from continuous machine-to-machine communications. Along with huge opportunities, this also poses problems: sensor measurements may conflict, processing times of logical and statistical reasoning algorithms increase non-deterministically polynomially or even exponentially, and wireless networks might become congested by the transmissions of all measurements. Bayesian networks are a good starting point for inference algorithms in pervasive computing, but still suffer from information overload in terms of network load and computation time. Thus, this work proposes to distribute processing with a modular Bayesian approach, thereby segmenting complex Bayesian networks. The introduced "Bayeslets" can be used to transmit and process only information which is valuable for its receiver. Two methods to measure the worth of information for the purpose of segmentation are presented and evaluated. As an example for a context-aware service, they are applied to a scenario from cooperative vehicular services, namely adaptive cruise control.

*Keywords:* Vehicle-to-Vehicle, Information Dissemination, Information Assessment, Bayeslet, Bayesian Network, Context Inference, Machine-to-Machine Communication

---

## 1. Introduction

The added value of ubiquitous computing systems stems from the fact that participating devices have access not only to their own information, but also to the information

---

*Email addresses:* [korbinian.frank@dlr.de](mailto:korbinian.frank@dlr.de) (Korbinian Frank), [matthias.roeckl@in2soft.de](mailto:matthias.roeckl@in2soft.de) (Matthias Röckl), [t.pfeifer@computer.org](mailto:t.pfeifer@computer.org) (Tom Pfeifer), [patrick.robertson@dlr.de](mailto:patrick.robertson@dlr.de) (Patrick Robertson)

of other relevant service providers, sensors or information consumers. The information received from heterogeneous and redundant sources is usually evaluated and fused with one's own information to form new, more accurate, or more reliable knowledge [33]. This *inference* on the available information is necessary for complex situation estimations e.g. to avoid collisions in road transportation.

Information is exchanged via centralised architectures like the internet, but also in ad-hoc networks like *vehicle-to-vehicle (V2V)* communication networks in safety-critical driver assistance systems [38] (cf. Figure 1) or communicating personal smart spaces [10]. In particular the ad-hoc networks in the latter situations are always wireless, may exist only for some moments and their capacity is limited. Depending on the current situation, sensor measurements can have a high importance to a node and its neighbours (e.g. to prevent a collision). However, the transmission of information over the wireless channel consumes bandwidth and thus prevents others from transmitting their – maybe more important – information. Whether information shall be exchanged hence becomes an information-theoretic problem which requires decision-making based on the information which is to be disseminated and the costs implied by the dissemination. The algorithms of section 4 introduce concepts for adaptive information dissemination using a utility-based decision-making approach.

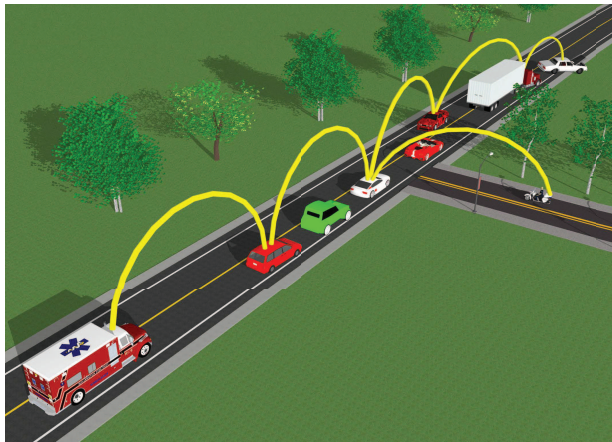


Figure 1: Vehicle-to-vehicle communication helps to avoid collisions in unclear situations [37].

The information that would be available to be exchanged in such mobile and highly dynamic environments exceeds the network capacity. Further to network bandwidth being a critical resource, also the processing capacity of many mobile computers would not be able to cope with an overload of perhaps even redundant information. Imagine the following standard example of ubiquitous computing, from the domain of Unified Messaging: Calls shall only be forwarded to the recipient if he is available, i.e. in a position to answer the phone. This depends on his agenda, his location, his current activity, time, the persons in his surroundings, ambient context (e.g. noise level) and also the caller and his status (e.g. calls in emergency situations should be forwarded). In addition, most of the mentioned input context information depends again on many other context sources and sensors. All this information has to be collected and evaluated on

the resource constrained mobile phone to infer the availability. The powerful methods for Bayesian networks are a good starting point for realising such context inference. They will be the basis for the approach presented in this paper. The NP-hard complexity [9] of probabilistic inference however entails the need for dedicated algorithms which are able to cope with high amounts of information.

The problem of information overload hence has to be solved, to reduce both, network load and computation time. Our general idea for this solution is the introduction of mechanisms where information is assessed before transmission and processing. The presentation of such mechanisms is the objective of this paper, i.e. rather the contribution to solving the logical problem in computability theory than the tuning of performance. We propose the Bayeslet concept to segment Bayesian networks, allowing for distributed inference of the situation and reduced transmission needs, as well as the “rating” of information which can automate the decision of which information needs to be transmitted for an inference target. We demonstrate the applicability in cooperative driving assistance using V2V networks as a special case of context inference where it reduces the network traffic and computational burden for processors.

After a review of the background theory and the related work in the following section, section 3 introduces the Bayeslet concept, a way to bundle information for local value determination and evaluation. Section 4 describes the approaches to define the worth of information which are then applied to an example from collision avoidance in vehicle networks in section 5. Finally, we close this paper with a conclusion and an outlook on how we will pursue and integrate this work in the future.

## 2. Background Theory and and Related Work

### 2.1. Fundamentals of Bayesian Networks

Often *Bayesian Networks (BNs)* are used for information processing and situation estimation [41]. They provide the necessary freedom in modelling situations for inference as well as the adaptability to new situations by automated learning from data. A BN consists of a set of *Random Variables (RVs)*  $V = \{A_1, A_2, \dots, A_n\}$ , a set of dependencies  $E = \{(A_i, A_j) | i \neq j, A_i, A_j \in V\}$  between these RVs and a *joint probability distribution (JPD)*  $P(V) = P(A_1 \cap A_2 \cap \dots \cap A_n)$ .  $P$  is the product of the *Conditional Probability Distribution (CPD)* of every RV  $P(A_i) \forall A_i \in V$ . A BN must not contain directed cycles. This model subsumes a great variety of other stochastic methods, such as Hidden Markov Models or stochastic dynamic systems [2].

An important concept in BNs is *d-Separation* with the “d” standing for dependence. It helps to reduce the network to relevant parts for given observations. If two variables are d-separated relative to a set of RVs  $Z$ , then they are independent conditional on  $Z$  in all probability distributions of its BN. Taking advantage of d-separation constellations, a BN exploits the conditional independence to represent a JPD more compactly [32]. The minimal set of nodes that d-separates  $A$  from the rest of the network is called *Markov Boundary* [30]. Furthermore BNs are able to deal with missing or uncertain data (arising from stochastic sensors or via uncertain data links) and can be built or modified either by machine learning algorithms or by human expertise.

A particular class of BNs are *Causal Networks* [31], where dependencies are interpreted as causal influence in the real world. Despite the fact that the dependency is

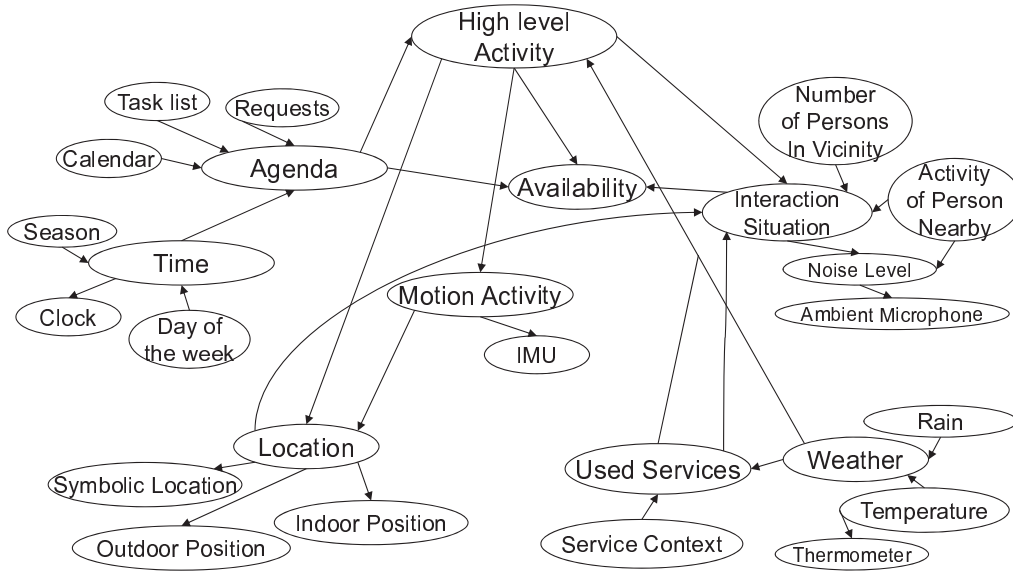


Figure 2: A grossly simplified Bayesian network representing important contextual aspects of a pervasive system user. Her high level context is influenced by environmental conditions, location, agenda, the presence and the interaction level with other persons, as well as her motion and behaviour. All these input factors again depend on other information or sensors, again represented as random variables.

directed, it allows for inference in both directions, inductive and deductive. Two examples for BNs can be found in Figures 2 and 3. While Figure 2 shows how complex and confusing already a strong simplification of a modelled snapshot can get, Figure 3 allows to understand the causal modelling.

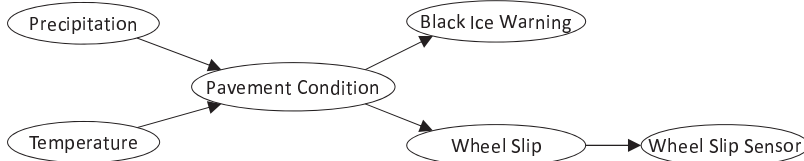


Figure 3: This basic version of a Bayesian network shows a small part of an inference network as used in a car to assess the danger of the current driving situation. The *PavementCondition* (which can be *dry*, *wet*, or *icy*) is caused by *Precipitation* and *Temperature*. The pavement condition itself causes *BlackIceWarning* in the news and *WheelSlip* which causes reactions of a *WheelSlipSensor*.

BNs have also been presented for use in context reasoning, e.g. in [5, 14, 15]. Using causal networks in context modelling and inference, every random variable represents a contextual entity (e.g. context attribute, sensor or service) and its values the states in which the entity can be. The structure and the probabilities in the CPDs encode the existing knowledge about a certain context. This can come from human expertise or from statistical analysis of contextual knowledge bases. Context inference is the process of generating knowledge for at least one of the context entities for which no knowledge is

available, in terms of Figure 2 for instance the level of a person’s availability depending on the knowledge about the current time and his current activity. With Bayesian networks this process is performed by probabilistic inference, which assigns the RVs in question the marginalised and normalised probabilities for all its values.

Just like for other inference approaches like logics (the satisfiability problem, SAT, is NP complete), the exact evaluation of large problems is intractable for Bayesian algorithms: inference in static Bayesian Networks is NP hard in the number of random variables [8]. The number of nodes, number of edges, and number of values per node are the factors which determine the evaluation time.

## 2.2. Related Work for Modular Bayesian Networks

Some approaches for modular Bayesian networks have been presented in the literature. They can be categorised as follows:

1. Different modules are composed to a common structure for inference.
2. A complete BN is used to set up a separated inference structure in which independent and separate inference is possible.
3. Completely independent and distributed inference is realised.

Examples for the first category with a common inference structure are the network fragments of Laskey and Mahoney [23] or also Paskin’s and Guestrin’s approach to modular Bayesian networks [27]. With this approach a BN can be adapted to the needs of a specific situation and under certain conditions reduced in size. However, distributed inference is not supported, all knowledge has to be aggregated on one computer. No transmission savings are realised.

The second category is used in Xiang’s *Multiply Sectioned Bayesian Networks (MS-BNs)* [46], and consequently also in the *Object Oriented Bayesian Network (OOBN)* [3, 4, 22] approaches and in Pavlin’s and de Oude’s modular BNs [28] which build upon MSBNs. Also Hwang’s modular Bayesian networks [20] represent the second category. Inference duration is reduced by building a common secondary inference structure of the different modules (like a join tree, [21]) in which different modules can be evaluated in parallel. However it does not make the different modules independent from each other and the system has to have knowledge of all existing connections between the modules. These approaches are not suitable for highly dynamic systems as in Ubiquitous Computing, as the construction of the secondary structure is more expensive than for normal BNs.

The requirements of large scale, highly dynamic environments are fulfilled in general by the third category, which however restricts the inference approaches to provide approximations of the searched posterior probabilities. The inference approaches of Hwang’s modular BNs [19] or the hierarchical and hybrid BNs [45] could realise this, but are not ideal implementations of this idea. Both approaches only allow for static modelling and do not optimise the network and dependency design, but use knowledge of the overall BN to distribute information into different modules in the design phase. Dynamic construction of networks of BN modules is not considered.

## 2.3. Basics of Probabilistic Information Assessment

The worth of any information depends on the situation. Functions are needed to compute this worth. In general they are called *value functions*. These are often treated

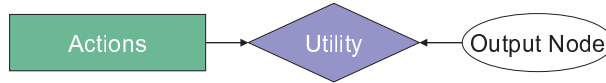


Figure 4: A *Decision Network* with one decision node, one utility node and one standard node representing a random variable.

in the framework of utility theory, e.g. in [24]. Functions suitable for value function should be e.g. never negative [21]. Further, it is desirable that some information gives certainty, i.e. the respective value function should assign such information (probabilities close to 1 or 0) high values in contrast to uncertain information (close to uniformly distributed RVs, e.g. probabilities around 50 % for binary RVs). Examples could be the classic entropy function as defined by Shannon [43], or the variance function.

Moreover, in the field of probabilistic information retrieval, probabilities are computed that a document is relevant for a given query. Hence, also the relevancy assessment turns into a problem of probabilistic inference. The underlying model thereby determines the relevancy decision: The Probabilistic relevance model [35] is often used with the Okapi BM25 [36] relevancy model for the relevancy assessment of documents. The Latent Dirichlet Allocation was first presented as a graphical model for topic discovery by David Blei et al. in [6]. Many of these models can be summarized as feature-based retrieval models [26] where relevancy is subdivided in features (e.g. relative frequency of search terms) and these are combined to an overall value function.

A more elegant way however is the direct usage of the functionalities offered by Bayesian networks. An extension of them allows for the direct inclusion of value functions in the so called *decision networks* or *influence diagrams* [21]. Decision networks introduce two new types of nodes, *decision nodes* and *utility* or *value nodes*. Utility nodes represent a utility or cost function in which every combination of events can be assigned a value. They cannot have states nor RVs as child nodes and are modelled as child nodes of *decision nodes* whose values represent the possible actions. They are imposed from the outside to meet some optimisation objective [30]. Within a decision network, decision variables are typically depicted as rectangular boxes and value functions as diamond-shaped boxes as shown in Figure 4.

#### 2.4. Conclusion and Requirements

Bayesian networks and methods for information assessment provide the necessary tools to infer context in ubiquitous computing environments and to determine the *utility* of information. The approaches for modular networks are however not directly applicable, as they always infer the result based on all the information available. It is necessary however to reduce this amount of information and to infer only upon the most relevant information. The necessary approach has to support the following features:

- (1) Decentralised, modular inference:  
To reduce load on servers, information coming from remote devices or smart environments should be preprocessed at their origin as far as possible. Therefore inference has to be modular, encapsulating context from different environments and enabling to plug the different modules together for final inference.

- (2) Reduction to relevant input:  
To shorten inference time, the amount of low level context taken into account has to be reduced to a minimum. Only situational information relevant for the respective high level context must be taken into account. Modules that are not relevant can be neglected.
- (3) Support for different characteristics of context, different inference algorithms:  
Not all context will be stemming from Bayesian inference approaches, but should still be incorporated in high level context inference, as far as possible. This can be realised with different inference modules using different inference methods.
- (4) Efficient adaptability to changes in the dynamic environment:  
As soon as previously disregarded information becomes relevant to the system it should be considered. Thus, due to the dynamic nature of context-aware computing systems, inference has to be adaptable to changes, such as the inclusion of new sensors or new contextual aspects.

### 3. Bayeslets: Segmentation of Bayesian Networks

Bayesian networks representing the full knowledge of all known influence to a certain fact can be immense. An example cited very often is the BN of the Computer-based Patient Case Study (CPCS) described by Pradhan et al. in [34]. This multiply connected network covering knowledge about internal medicine consists of 448 multi-valued nodes and 906 links. The provision of an inference result which would allow for a fast medical diagnosis is not possible with such a network.

The situation of a user of a ubiquitous computing system can encompass a very large range of human activities, sensor readings, system usage information, as well as such data relating to other people – all represented as random variables in a very large BN. As demonstrated by the author in detail in [11] it is impossible to include all such RVs in a representation used in inference, as the resulting BN would be too large to be processable. The example from Figure 2 is only a gross simplification where some nodes already aggregate the information of complete sensor BNs, and where only a small subset of information relevant for a user is represented, ignoring its relation to all other users, such as family, friends or colleagues, as this would result in networks with several thousands or millions of RVs.

We can conclude that a complete Bayesian network, modelling all possible influence factors, would be intractable for the high number of nodes and the resulting computational complexity. Therefore this work proposes to segment all this information into thematic groups called *Bayeslets* [11]. This word has been composed of “Bayes”, as Bayeslets represent knowledge in the way of Bayesian networks, and “let” suggesting (like in the term “Applet”) that they are small, mobile and self-contained modules, executable wherever needed.

#### 3.1. Specification of Bayeslets

The properties defining a Bayeslet are the following:

*Owning Entity.* The owning entity is the identifier of the Bayeslet’s owner in the ubiquitous computing system. It is necessary for privacy and unique identification.

*Output Context Attributes.* The output context attributes are the random variables inferable with a particular Bayeslet. Together with their Markov boundary, they constitute the nodes contained in the Bayeslet.

*Input Context Attributes.* The input nodes specify the observed nodes in the inference rule. It is only these nodes into which other Bayeslets can be plugged. Only those Bayeslets can be plugged which provide the respective context attribute as output context attribute. Output and input context attributes constitute the *interface* of a Bayeslet. The example in Figure 5 shows how input and output nodes are represented differently.

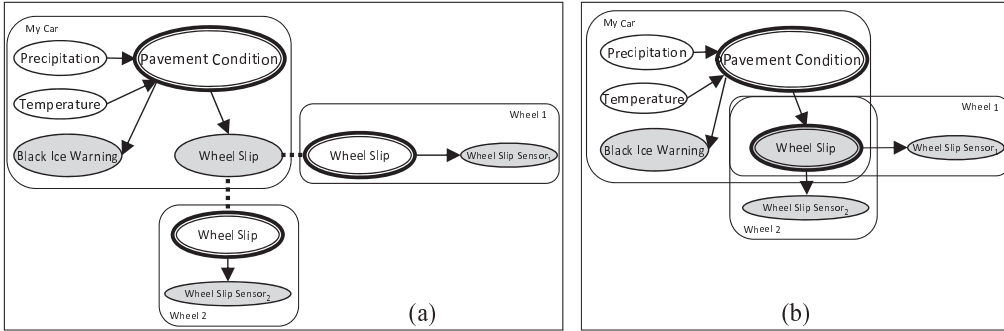


Figure 5: Connection of Bayeslets: the Bayeslet for the inference of the *PavementCondition* is connected to two Bayeslets determining the *WheelSlip* incorporating a *WheelSlipSensor*. Input nodes are shaded in grey, output nodes have a double border.

*Context Inference Rule.* The field for this property specifies the actual inference rule module. For static discrete BNs it consists of the nodes with their states and the respective CPDs which implicitly define the BN structure. According to requirement (3) from section 2.4 however, also other contextual rules have to be supported, e.g. A-boxes and T-boxes from Description Logics [17], mathematical rules between values of RVs or dynamic Bayesian networks. Hence the contents of this field depend on a used context model and the inference algorithm.

For Bayeslets which are not evaluated locally, because they are related to remote information or information of a different user, no context inference rule is specified in this field, but a link to a remote access point providing the inference result.

*Inference Algorithm.* The field for this property specifies the algorithm used to evaluate the inference rule, specified in the *Context Inference Rule* field. This is necessary, as the Bayeslet concept allows other inference algorithms beyond Bayesian, in order to be able to cope with the heterogeneity of inference problems. The only requirement for the required algorithm is the provision of a posterior distribution for the output context attributes. Examples are the algorithm *Probability Propagation in Trees of Clusters (PPTC)* [18] for a static BN as inference rule, *C.45* [44] for a decision tree as inference rule or a Kalman filter for a suitable dynamic Bayesian network.



*Meta information.* A context inference system based on Bayeslets requires further information about the Bayeslets to increase its efficiency. Examples for such meta information are the type of the value ranges (numerical, ordered, hierarchical, etc.), the expected posterior probability of the output nodes, or the costs caused, depending on size, inference duration and inference frequency.

### 3.2. Guidelines for Bayeslet Modelling

Using Bayeslets, the BN from Figure 2 can be grouped into several subdomains shown in Figure 6. Thereby, only those Bayeslets which are necessary for an inference request have to be connected, together with relevant information of other users which can differ from situation to situation.

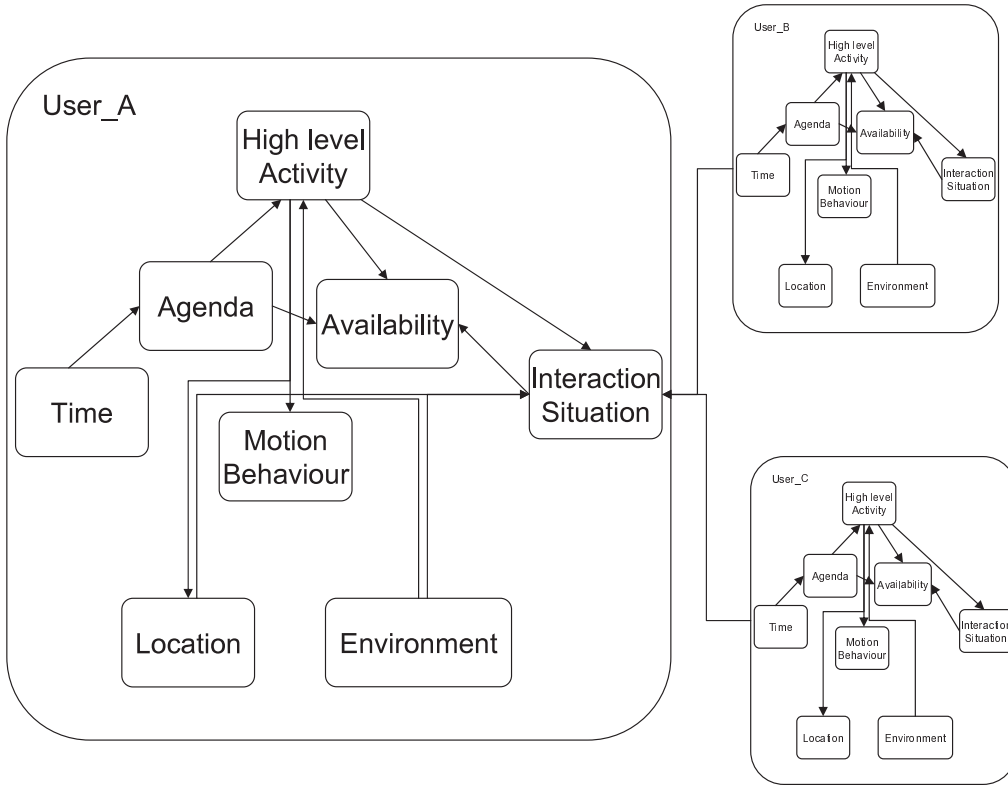


Figure 6: Inference network based on Bayeslets. The main components from Figure 2 are grouped into Bayeslets, shown in rectangles with rounded corners. The composition of those smaller Bayeslets forms Bayeslets per user which contain only the relevant inner Bayeslets.

There are some general guidelines on how and where to separate all available random variables into Bayeslets, i.e. how to define the borders of domain knowledge.

1. An input node forms a separate Bayeslet together with the nodes influenced. The simplest case can be seen in Figure 5, where only one input node and the measured random variable form a Bayeslet. This rule can be specified for sensors as input

nodes. Both the number and the selection of sensors are not crucial for the inference target. It is desirable that sensors can be added or removed and that no sensor is used if not necessary and that several sensors can be used if this adds value. Hence they should be individual Bayeslets.

2. A Bayeslet only contains information about one user, the owner of the Bayeslet. Information about other users is plugged in when necessary. The big advantage of this approach is that not all information of possibly relevant people has to be modelled and inferred upon, but only of the currently relevant people.
3. Unconnected parts of BNs can be separated into different Bayeslets. Obviously, if there is no path (ignoring edge directions) between two random variables of a Bayesian network, they are independent and do not mutually impact on the inference of the other RV.
4. Parts of the overall information that would be d-separated [29] by an evidence node in a monolithic BN can be divided into Bayeslets. If there is a node (set) directly connected to a sensor or always available information, as it is specified for dynamic Bayesian networks, it can be regarded as always observed. If this node (set) d-separates two parts of the network, these can be separated in different Bayeslets.
5. Weak edges, i.e. those representing (almost) uniform distribution can be disregarded, if this enables a separation. Almost uniform distributions are those where there is only a small standard deviation among all single likelihoods in the CPD. Removing edges can disconnect graphs, if the edge had been a *bridge*. A bridge is the only connection (ignoring edge directions) between two subgraphs. After its removal, both subgraphs are independent, as no child-parent relation between any two nodes from the different subgraphs exists. Removing edges that have not been bridges still influences the d-separation of the formerly connected nodes and therefore fulfill the conditions of point 4.
6. A maximum number of nodes per Bayeslet can be defined. A number of 20 – 30 nodes has shown to be tractable on mobile devices and easy enough to manage. Cuts should be made by removing a minimum number of edges, hence preferably bridges, preferring weak links and such that enable d-separation. Furthermore, assuming evidence in nodes which are not in reality marked as evidence nodes, enables d-separation and can be used to force a separation into several Bayeslets.

Weakness of edges thereby can be determined by different means, e.g. Mutual Information of the linked random variables [43], or the deviation of the child’s CPD from a uniform distribution.

#### 4. Composition of Bayeslets

The application of the Bayeslet concept requires the introduction of an additional step before the actual context inference process: single Bayeslets have to be assembled to the BN eventually evaluated. In order to allow for efficient and precise Bayesian context inference taking into account the available information at request time, the concept of *Dynamic Composition of Bayeslets* has been introduced by the authors in [12, 13]. To save transmission costs and inference time, the relevant Bayeslets have to be selected. As the

relevancy depends on the current situation and the requested output node, composition has to be a dynamic process at run-time of the context aware computing framework.

Prerequisite for such dynamic assembly is a database of the available Bayeslets with the information necessary to compute costs and information gain. This database stores all local Bayeslets. Information about remote Bayeslets enters the database by advertisements when remote smart spaces meet. The advertisements are not fully fledged Bayeslets, but only stubs representing the Bayeslet header and an access point for remote evaluation.

For joining two Bayeslets  $\mathbf{X}$  and  $\mathbf{Y}$ , mechanisms to find pluggable Bayeslet candidates in the database are necessary. Plugging candidates are such Bayeslets where an input node of  $\mathbf{X}$  matches an output node of  $\mathbf{Y}$ .

#### 4.1. Dynamic Bayeslet Assembly Process

The BN composed of Bayeslets has to be minimal, but should not disregard important links. Moreover, assembly of Bayeslets poses challenges, as it is recursive, because linked-in Bayeslets can allow or require input from other Bayeslets. This introduces two main risks:

- **Loops:** If a Bayeslet Network, i.e. the directed graph consisting of Bayeslets and their directed connections, would result in having a loop, also the complete BN formed by this graph could contain a directed loop, which is forbidden in BNs. Algorithms for deadlock prevention or breadth-first search can be applied to exclude such cases every time a new Bayeslet is to be added.
- **Excessive Linking:** if all available interface nodes in Bayeslets are linked and joined with other Bayeslets, at least in a large scale pervasive computing framework this would result in a very large network, possibly until all Bayeslets in the knowledge base are linked. So the same, complete inference BN that was referred to in section 3 would be recreated. Its evaluation could lead to unacceptably high delays.

To answer an inference request, the Bayeslet with the target (output) node is selected and has to be assembled with Bayeslets linked into the input nodes. The selection of such Bayeslets has to be performed in a new step, preceding the inference itself as shown in Figure 7. Decision criterion has to be the costs added by joining the network weighted against the information gain. Costs are mainly transmission and evaluation costs of a Bayeslet, depending on the size and the communication costs. The information gain is determined by the expected impact on the posterior probability of the output node. Only those Bayeslets exceeding a certain threshold of added value will be joined.

#### 4.2. Composition Criteria

To decide whether to connect two Bayeslets, the utility of the additional information has to be determined. In utility theory the term *utility* is defined by the value a piece of information provides to the system.

The utility of an additional piece of evidence  $y \in Y$  can be quantified by the difference between the utility with the additional evidence and without the additional evidence. Here,  $y$  refers to the evidence of a random variable (e.g. a measurement of an accelerometer) which is not independent of another random variable  $X$  (e.g. the activity of the

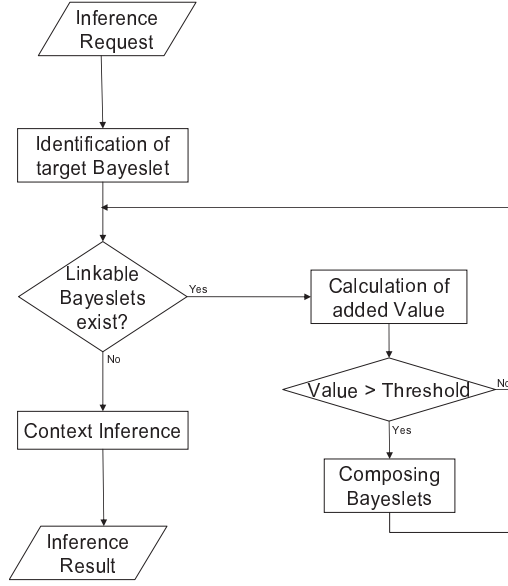


Figure 7: The process of composing Bayeslets to a complete inference rule.

person). Thus, the utility of evidence  $y$  shall be determined by its impact on  $X$ . This impact is defined by:

$$U(X : y) = \underbrace{U(X|y)}_{\text{Utility of } X \text{ given known } y} - \underbrace{U(X)}_{\text{Utility of } X \text{ not given } y}, \quad (1)$$

with  $U(X) : X \rightarrow \mathbb{R}$  being the utility function which maps a random variable  $X$  to a real number.  $U(X|y)$  is the utility function that maps the random variable  $X$  given evidence  $y$  to a real number. Since  $X$  is subject to uncertainty, Eq. (1) can be enhanced to calculate the *expected utility (EU)* gain by summing over all states weighted by their probability of occurrence:

$$\begin{aligned} EU(X : y) &= EU(X|y) && - EU(X) && (2) \\ &= \underbrace{\sum_{x \in X} U(x|y)P(x|y)}_{\text{Expected Utility of } X \text{ given known } y} && - \underbrace{\sum_{x \in X} U(x)P(x)}_{\text{Expected Utility of } X \text{ not given any } y} && . \end{aligned}$$

However, it is not possible to determine  $y$  unless the Bayeslets are connected. The solution is to calculate the expected utility which can be gained from acquiring any  $y \in Y$  instead. Therefore, one has to sum over the utility  $U(x|y)$  of all possible outcomes

of  $y$ , weighted not merely by  $P(x|y)$  but also by  $P(y)$ , the prior probability of  $y$ :

$$EU(X : Y) = EU(X|Y) - EU(X) = \underbrace{\sum_{x \in X} \sum_{y \in Y} U(x|y)P(x|y)P(y)}_{\text{Expected Utility of } X \text{ given unknown } Y} - \underbrace{\sum_{x \in X} U(x)P(x)}_{\text{Expected Utility of } X \text{ not given any } y} . \quad (3)$$

The equations above are defined in utility theory to rate the importance of a node  $Y$  for  $X$  with a known direction of their dependency. Bayeslets however represent sets of nodes calculating a posterior probability which is introduced as soft evidence to the input node of another Bayeslet. The substitute structure of *virtual evidence* for soft evidence [41] allows the direct application of the concepts above: Soft evidence is mapped to the conditional probabilities of a virtual child node with hard evidence. The expected posterior distribution of Bayeslets is static and therefore maintained in the meta information of the Bayeslet specification. Hence we can always assume that the plugged-in Bayeslet is represented by a child node  $Y$  of an input node  $Y_X$  in the Bayeslet with the queried output node  $X$ .

To account for already acquired knowledge, e.g. from already connected Bayeslets, Eq. (3) can be enhanced to consider contextual knowledge  $c$ :

$$EU(X : Y|c) = \underbrace{\sum_{x \in X} \sum_{y \in Y} U(x|y, c)P(x|y, c)P(y, c)}_{\text{Expected Utility of } X \text{ given unknown } Y \text{ and known } c} - \underbrace{\sum_{x \in X} U(x|c)P(x|c)}_{\text{Expected Utility of } X \text{ given known } c} . \quad (4)$$

The term  $EU(X : Y|c)$  determines the utility which can be expected by the connection of a Bayeslet represented by  $Y$  given already acquired context  $c$  to input node  $X$ .

In the following subsections, it will be shown that  $EU(X : Y|c) \geq 0$  for the proposed utility functions. This means that additional evidence never decreases the utility and, thus, Bayeslets in most cases shall be connected. Therefore also the costs have to be taken into account which can arise from e.g. higher processing load, communication costs in case Bayeslets reside on remote entities or monetary costs if evidence from a remote Bayeslet is provided by a commercial service, e.g. a commercial weather service for highly accurate weather information.

To take into account costs that reduce the gross utility, Eq. (4) can be enhanced to the so called *Net Expected Utility (NetEU)*:

$$NetEU(X : Y|c) = EU(X : Y|c) - C(Y) , \quad (5)$$

in case costs for all  $y \in Y$  are equal. If this is not the case, the  $NetEU(X : Y|c)$  is calculated by:

$$NetEU(X : Y|c) = \sum_{x \in X} \sum_{y \in Y} (U(x|y, c) - C(y))P(x|y, c)P(y, c) - \sum_{x \in X} U(x|c)P(x|c) . \quad (6)$$

In the following, two different kinds of utility functions are proposed which can be used as criteria for the connection of Bayeslets similar to the concepts of information gathering and dissemination by Röckl in [40].

#### 4.2.1. Probability-based Utility Functions

Often the utility of a random variable increases the “better it is known”, i.e. the less the inherent uncertainty is. Also humans regularly acquire new information from independent sources if they are uncertain about the “true” state of an unknown process until they feel confident enough: If we are uncertain about the weather tomorrow, we check the recent weather forecast.

A suitable utility function would increase (for instance logarithmic), if new information adds value. The binary logarithm of the probability as used by Shannon in [43] fulfils the requirements. In this case  $EU(X : Y)$  of Eq. (3) is equivalent to Shannon’s Mutual Information  $I(X : Y)$ :

$$\begin{aligned} EU(X : Y) \equiv I(X : Y) &= \sum_{x \in X} \sum_{y \in Y} \log_2 P(x|y)P(x|y)P(y) - \sum_{x \in X} \log_2 P(x)P(x) \\ &= -H(X|Y) + H(X) \quad [in\ bits] , \end{aligned} \quad (7)$$

with  $H(X)$  being the entropy of the random variable  $X$ ,  $H(X|Y)$  being the conditional entropy of  $X$  given  $Y$ .

In this case,  $EU(X : Y) \geq 0$  because:

$$\begin{aligned} EU(X : Y) &= - \sum_{x \in X} \sum_{y \in Y} P(x, y) \log \frac{P(x)}{P(x|y)} \quad (8) \\ &\geq^{(*)} - \log \sum_{x \in X} \sum_{y \in Y} P(x, y) \frac{P(x)}{P(x|y)} \\ &= - \log \sum_{x \in X} \sum_{y \in Y} P(x, y) \frac{P(x)P(y)}{P(x, y)} \\ &= - \log \sum_{x \in X} \sum_{y \in Y} P(x)P(y) = - \log 1 = 0 . \end{aligned}$$

As  $(-\log)$  is a convex function [7], in the line in Eq. (8) marked  $(*)$ , the Jensen inequality  $E_x[f(x)] \geq f(E_x[x])$  for convex functions  $f$  [25] has been used.

To compute the net expected utility, a measure for the costs has to be specified. It can include memory consumption, communication, and CPU costs, depending on the needs of the implementing system.

Regarding CPU costs, i.e. inference time, the following could be an adequate measure:

$$C_{CPU}(Y) = H(Y) - \frac{H(Y)}{n_Y^a + b} , \quad (9)$$

with the entropy of  $Y$ ,  $H(Y)$ ,  $a \in \mathbb{R}$ ,  $b \geq 1$  constants and  $n_Y$  the size of the biggest cluster in the junction tree [18] of the Bayeslet containing  $Y$ .  $C_{CPU}(Y)$  has the entropy of  $Y$  as an upper bound, to remain comparable with the mutual information. If the maximum cluster in the junction tree (which has large influence on the evaluation complexity, see [16]) is very small, the denominator of the second summand  $n_Y^a + b \approx 1$  and the costs  $C_{CPU}(Y) \approx 0$ .

For transmission and memory costs  $C_{TM}(Y)$ , the size of the rule and its representation are the relevant factors.  $n_Y$  from Eq. 9 could be replaced by the number of bits

necessary for a serialized representation. Requirements for a Bayeslet representation language have been discussed by the authors in [14]. Battery costs are influenced both by CPU costs and transmission costs, hence should be represented as a weighted sum of the individual costs. This also holds for the overall costs  $C(Y)$ :

$$C(Y) = \sum_i a_i \times C_i(Y), \quad (10)$$

where  $\sum_i a_i = 1$  and  $C_i$  is representing an individual cost function like  $C_{CPU}$  or  $C_{TxM}$ . With this approach, the comparability to the mutual information can be maintained. The costs should be static and can be stored as a constant in the Bayeslets meta information field.

#### 4.2.2. Decision-based Utility Functions

Although normally it is beneficial to head towards less uncertainty in the higher level information, in decision support systems utility shall be based on the outcome of actions instead of uncertainty reduction. Therefore the Bayeslet model can be extended by the notion of *action*, known from decision theory. Formally, Savage defines an action as “a function attaching a consequence to each state of the world” [42]:

$$\begin{aligned} a : X &\rightarrow X \\ a(x) &= x_a, \quad \text{with } x, x_a \in X, \end{aligned} \quad (11)$$

with  $x_a$  being the consequence of  $x$  after performing the action  $a$ .

To incorporate actions into Bayesian networks, decision networks can be used. A rational decision maker will choose the action which maximises the resulting utility. Thus, the expected utility can be replaced by the *maximum expected utility (MEU)* [41]:

$$EU(X|Y) \equiv MEU(X|Y) = \arg \max_{a \in A} EU(X|Y, a) = \arg \max_{a \in A} \sum_{x \in X} U(x|Y, a)P(x|Y). \quad (12)$$

If  $EU(X|Y)$  is substituted by  $MEU(X|Y)$  in Eq. (3),  $EU(X : Y)$  is equivalent to the so called *Value of Information (VoI)* in information value theory [21]:

$$EU(X : Y) \equiv VoI(X : Y) = MEU(X|Y) - MEU(X). \quad (13)$$

Intuitively understandable, additional evidence never decreases the *VoI*, however  $VoI(X : Y) \geq 0$  can also be shown by the following proof:

$$\begin{aligned} MEU(X|Y) &= \sum_{y \in Y} P(y) MEU(X|y) \\ &= \sum_{y \in Y} P(y) \arg \max_{a \in A} \sum_{x \in X} P(x|y)U(x|y, a) \\ &\geq \arg \max_{a \in A} \sum_{x \in X} \sum_{y \in Y} P(y)P(x|y)U(x|a) \\ &= \arg \max_{a \in A} \sum_{x \in X} P(x)U(x|a) = MEU(X) \end{aligned}$$

Here, the conditional independence of the utility function  $U(x|y, a) = U(x|a)$  and the Jensen inequality  $E_x[f(x)] \geq f(E_x[x])$  [25] have been used.

### 4.3. Summary

Both approaches, probability- and decision-based utility determination, fit the requirements of dynamic composition. They use the expected evidence of a plugging candidate Bayeslet to measure the impact on either the output node itself or on actions relevant to the output node. The information necessary for this is static and therefore can be maintained in the Bayeslet specification and used without computation and communication overhead.

For the *VoI*, the design of the utility function and its harmonisation with the cost function for the composition is challenging and tedious, but adds flexibility to the system in comparison to pure uncertainty reduction. It would be the task of a human domain expert.

## 5. Application and Evaluation

The following subsections shall illustrate and verify both utility measures proposed in the preceding section with a common application. Thereby, we will focus rather on showing the solution of the computability problem than on a generic performance analysis.

### 5.1. Application Example

The application Cooperative Adaptive Cruise Control (CACC) [1, 39] is used to compare the behaviour of the two approaches to utility determination presented above. CACC is an applications in vehicle-to-vehicle communication for collision avoidance.

Hence, the vehicles are part of the ubiquitous computing environment. The communication channel has to be used collaboratively by all vehicles in the network and timely reception of messages is crucial for safety relevant applications like CACC. Therefore, bandwidth has to be shared intelligently. This implies in particular that irrelevant information should not be transferred. Applied to context inference, this means that a vehicle trying to infer its status should not request soft evidence from a remote Bayeslet which does not add significant value. Transmission over the wireless channel would consume bandwidth and thereby prevent others from transmitting their, maybe more important, information.

In the following a CACC application running in a car is considered. On a road with multiple lanes in the same direction, CACC has to be aware of the actual lane the preceding vehicle is driving on. If, for instance, the car is located on the centre lane of a three-laned road, the preceding vehicle can be located on the same, the right or the left lane. Context inference can infer this from the available information and provide this information to CACC [13].

A part of this situation is modelled in Figure 8. The three states of the possible location of the preceding vehicle are denoted in the following as values *centre*, *right* and *left* of the random variable *Lateral Distance (LD)*. Information sources for the lateral distance are the built-in radar system of the car itself, but also the messages with *Global Navigation Satellite System (GNSS)* information of the preceding vehicle received via V2V communications. Both information sources form Bayeslets which can be connected to the Bayeslet for the *Lateral Distance*, if their *NetEU*, cf. Eq. (5), is sufficiently high.



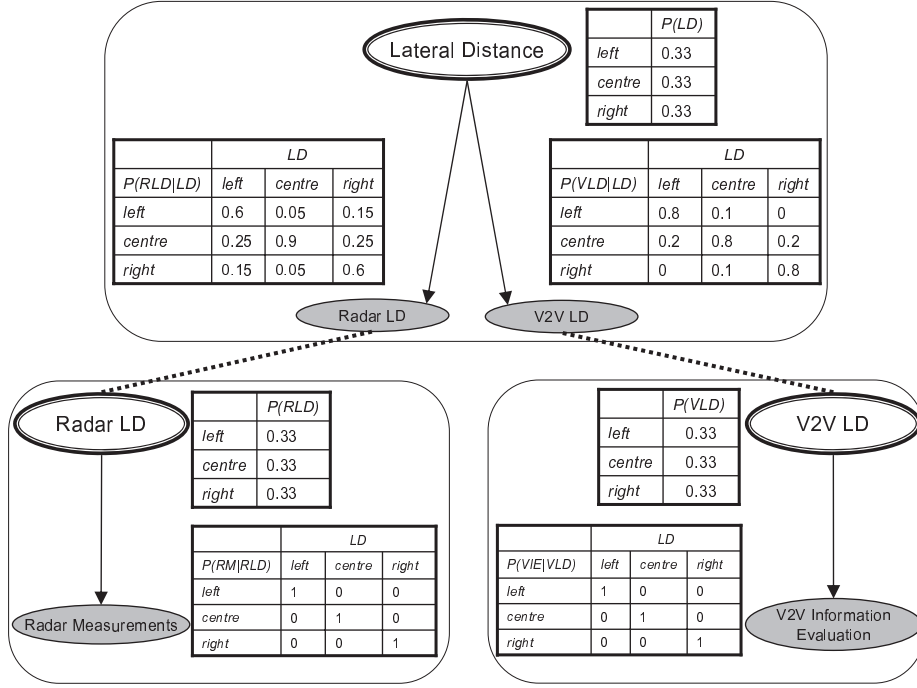


Figure 8: Three simplified Bayeslets to infer the *Lateral Distance* ( $LD$ ) of two cars for *Context Aware Adaptive Cruise Control*. The dotted lines indicate the possible connections of the Bayeslets, the boxes next to the random variables give their corresponding prior and conditional probability tables, respectively.

The Bayeslets used for this purpose are simplified to the minimal case so the concepts from section 4.2 can be illustrated best. In particular the Bayeslets of the information sources have been strongly simplified, so they have the same priors of their output nodes, the same size, and the same connections. Both information source Bayeslets hence have the same computational costs, so the evaluation can focus on the two proposed utility functions.

## 5.2. Mutual Information Based Composition Decision

Section 4.2 has presented the mutual information  $I(\mathbf{X} : \mathbf{Y})$  as an appropriate, probability based utility function to evaluate the relevancy of a Bayeslet  $\mathbf{Y}$  for the evaluation of Bayeslet  $\mathbf{X}$ . Thereby, the mutual information between two Bayeslets  $\mathbf{X}$  and  $\mathbf{Y}$  is reduced to calculating the mutual information between two random variables in the same Bayesian network – the output node of  $\mathbf{X}$  and the input node of  $\mathbf{X}$  with the prior probabilities of  $\mathbf{Y}$  as evidence.

The following evaluation concentrates on the behaviour of the mutual information between the three Bayeslets for *LateralDistance* ( $LD$ ), *RadarLateralDistance* ( $RLD$ ), and *V2VLateralDistance* ( $VLD$ ) shown in Figure 8, as the cost function for both input Bayeslets are the same, following the proposed cost function for computational costs from Eq. (9).  $H(RLD) = H(VLD) = 0.86$ , as the output variables of both Bayeslets have

three values with uniformly distributed prior probability. The same size of the Bayeslets results then in the identical value for the cost function, e.g.  $C(\mathbf{RLD}) = C(\mathbf{VLD}) = 0.77$ , if  $a = b = 1$ .

In the simplistic example used, Bayeslet  $\mathbf{LD}$  has to decide, if one or both of the Bayeslets  $\mathbf{RLD}$  and  $\mathbf{VLD}$  should be connected. Bayeslet  $\mathbf{LD}$  only contains input and output nodes, so the decision is made without incorporating any evidence:

$$I(\mathbf{LD} : \mathbf{RLD}) = 0.43 \tag{14}$$

$$I(\mathbf{LD} : \mathbf{VLD}) = 0.78 \tag{15}$$

Both Bayeslets hence add value to Bayeslet  $\mathbf{LD}$ . The value of Bayeslet  $\mathbf{VLD}$ , however, is expected to be higher, as the conditional probability distribution of node  $VLD$  expresses higher certainty and the CPD of  $RLD$  expresses lower sensor quality for the detection of *left* and *right*, e.g. due to reflections on the guard rails. Using Eq. (5) and taking into account that at this point the contextual knowledge  $c = \{\}$ :

$$NetEU(\mathbf{LD} : \mathbf{RLD}) = I(\mathbf{LD} : \mathbf{RLD}) - C(\mathbf{RLD}) = -0.34 \tag{16}$$

$$NetEU(\mathbf{LD} : \mathbf{VLD}) = I(\mathbf{LD} : \mathbf{VLD}) - C(\mathbf{VLD}) = 0.01$$

In this case only Bayeslet  $\mathbf{VLD}$  is connected, saving 50% of CPU time.

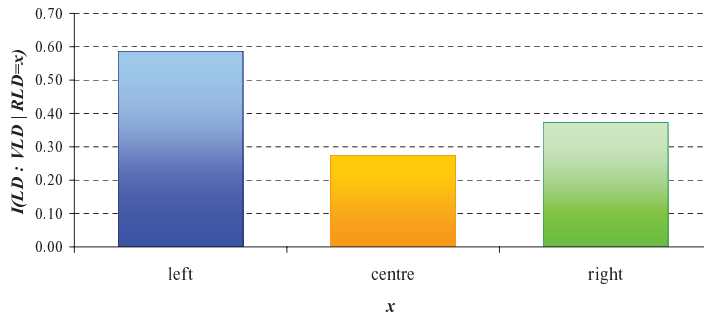


Figure 9: Evaluation of the mutual information between the Bayeslets  $LD$  and  $VLD$  given the different outcomes of Bayeslet  $RLD$ .

In the case of cars, the computing power is not as limited as for small devices. The critical resource in traffic is rather the network bandwidth. Taking this into account, context inference could decide in this case to always connect all local sensor systems like radar and to evaluate only the relevancy of external sources like the Bayeslet  $\mathbf{VLD}$ . Its relevancy depends on the values observed for the other Bayeslets.

Figure 9 shows the results for the mutual information between the Bayeslets  $\mathbf{LD}$  and  $\mathbf{VLD}$  given all possible observed values of Bayeslet  $\mathbf{RLD}$ . The mutual information varies between 0.59 and 0.27, as the utility of an additional Bayeslet depends on the probability of  $LD$  which has already been influenced by the already available evidence. The different degrees to which it still can be impacted by the additional Bayeslet determine the mutual information. Hence also in this case, given the costs of  $C(\mathbf{RLD}) = C(\mathbf{VLD}) = 0.77$ , no further Bayeslet is connected, independent from the value of  $\mathbf{VLD} \subseteq c$ , as the added value

is lower than the costs. In our simplistic scenario we would save all network transmission costs for the result of the remote Bayeslet.

The information dissemination with a probability-based utility function (sec. 4.2.1) has been implemented within a sequential Monte Carlo estimator (SMC) which is also known as particle filter (more details can be found in [38]). The detailed parameter specifications for the filter configuration can be found in Röckl et al. [40].

For the evaluation we implemented a time-triggered simulation environment which generates sensor output for simulated vehicles with 10 Hz update rate. The evaluation scenario used in this work is a straight and a “zig-zag” road with two vehicles following each other. Each vehicle is assumed to be equipped with a GNSS receiver, an odometer and a compass which provide erroneous measurements. The preceding vehicle sends out these measurements according to the algorithms defined in section 4.2.1.

Fig. 10 shows the expected utility  $EU(X : y)$  of the preceding vehicle during a simulation run on the “zig-zag” road. During the straight road segments the  $EU(X : y)$

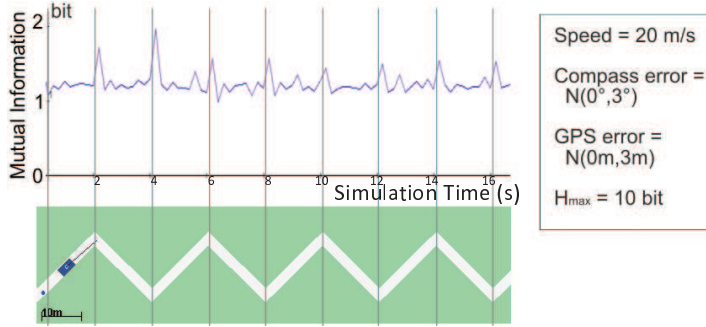


Figure 10: Mutual information on a “zig-zag” road with sharp 90° bends.

varies about 0.15 *bit* around 1.1 *bit*. Thus, there is no significant change and the message priority is low. After each sharp bend  $EU(X : y)$  shows distinguishable peaks since the prediction of the future position acts on the assumption of an ongoing straight movement with an unknown map. The update with recent evidence provides an unexpected change in this movement and thus has a high  $EU(X : y)$ . The peak height depends on the current measurement quality which is not shown in the figure.

Assuming now a threshold for the  $NetEU$  of 0.5 (cf. Figure 7), next to the same Bayeslet connection costs as above, then only values greater than the average are transmitted. This reduces the transmission frequency from the sensors’ 10 Hz to approximately 0.5 Hz – a reduction of 95% in transmission costs during the 17 seconds of this simulation.

Fig. 11 shows the progress of the  $EU(X : y)$  for the target vehicle driving on a straight road. Due to assumed faults in the positioning system, GNSS updates arrive more rarely with an update rate of 0.4 Hz. Every time a new measurement becomes available a peak in  $EU(X : y)$  can be recognized since the uncertainty of the prediction is significantly reduced by the new measurement. Up to now only the variation of  $EU(X : y)$  depending on the measurement value and its uncertainty has been inspected. In Fig. 12 the actual resulting update rate is evaluated when the  $EU(X : y)$  is used to decide whether to disseminate the measurement. If the measurement was retained, its  $EU(X : y)$  value



Figure 11: Mutual information on a straight road with GNSS position fixes every 2.5 s.

was added to the expected utility of the upcoming measurement resulting in a cumulative expected utility value. The costs for the dissemination were uniformly set to 10. Thus, every time the cumulated  $EU(X : y) > 10$  the message containing the evidence position, velocity and heading is disseminated.

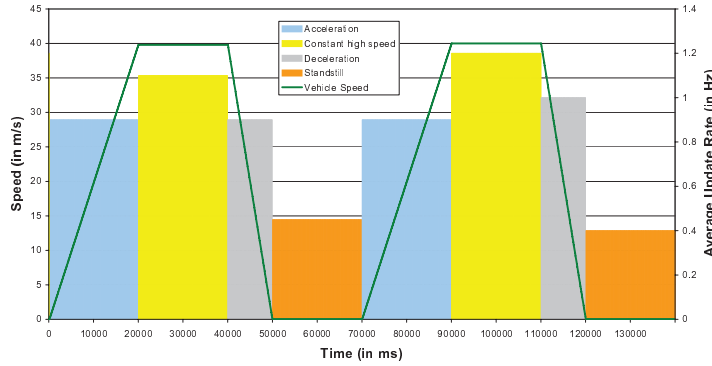


Figure 12: Average update rate during the driving manoeuvres *acceleration*, *constant high speed*, *deceleration* and *standstill*. Compared to the sensors' update rate of 10 Hz, reductions between 90 and 95 % can be achieved.

The continuous line depicts the velocity of the preceding vehicle. The vertical bars show the mean update rate during the acceleration, the constant high speed, the deceleration and the standstill phase. During the standstill the update rate is the lowest with about 0.4 Hz. During the acceleration and deceleration phase the update rate raises to 0.9 – 1 Hz and during the high speed movement the update rate reaches its maximum with about 1.1 Hz. This is justified due to the higher uncertainty of the movement prediction at higher speeds. Also here, compared to the sensors' update rate of 10 Hz, we can achieve reductions between 90 and 95 %.

### 5.3. Value of Information Based Composition Decision

The decision based utility function can be used to determine the (un-)certainty introduced by a Bayeslet as shown by the author in [11]. More importantly however, it

offers the possibility that more specific utility functions and decisions are defined. The following demonstrates the latter option, according to [13]. A tripartite utility function

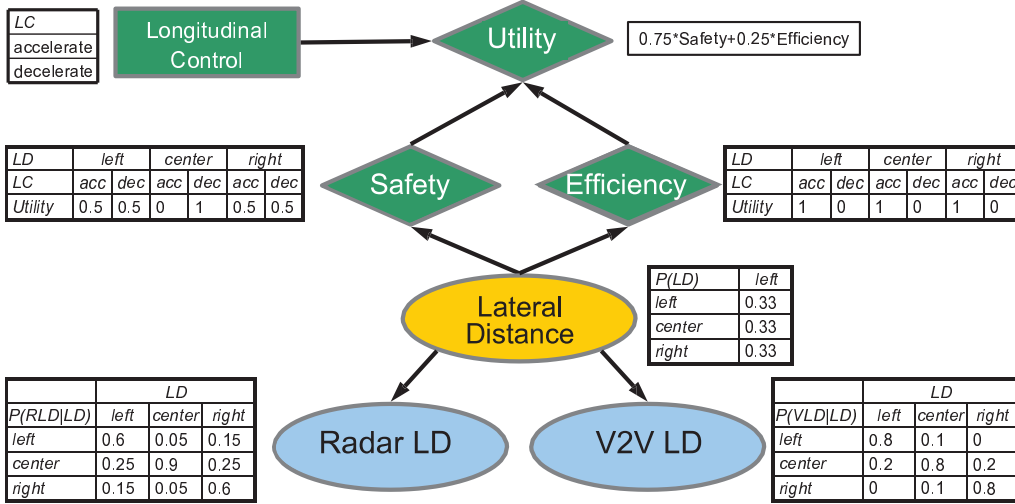


Figure 13: *Longitudinal Control* decision based on the *Lateral Distance* considering the information sources radar and V2V communication like in Figure 8. *Efficiency* and *Safety* measure the utilities as shown in the boxes next to their node representations. The overall utility influencing the decision is calculated by  $0.75 \cdot \text{Safety} + 0.25 \cdot \text{Efficiency}$ . Taken from [13].

has been specified. The basic utility functions measure the *Safety* and the *Efficiency* utility of an action. The overall utility function then represents the weighted sum of the two basic ones at the ratio of 3:1. The decision differentiates the two states *accelerate* and *decelerate*. The whole probabilistic decision network, extending the situation shown in Figure 8, is depicted in Figure 13. The explicit representation of the Bayeslets is neglected here as their outcome is introduced as evidence into the nodes *Radar LD* and *V2V LD* and the value of information can be computed only in this Bayeslet.

If no Bayeslet provides information, the state of *LD* is uniformly distributed, each state with a probability of  $1/3$ , see Figure 14 (left). In this case there is a tie between the actions *accelerate* and *decelerate*. Both have a utility of 0.5. In order to decide whether another Bayeslet including additional sensor measurements either from radar or V2V communication should be connected, the *VoI* is calculated. Figure 14 (left) shows that both Bayeslets yield  $VoI > 0$ , hence are relevant.  $VoI(RLD) = 0.11$  and thus carries more information than the Bayeslet for the information from V2V communications with  $VoI(VLD) = 0.10$ . These calculations are independent of the unknown actual state of the evidence and only use the prior probabilities of the Bayeslets' output nodes available from their specifications. Thus, the Bayeslet of the radar measurements is connected and evaluated.

When the radar Bayeslet is connected and has provided evidence to the input node *RLD*, the value of information and therefore the decision whether to connect the second Bayeslet is different. If, for instance, the radar has measured a *centre* state, the *VoI* of *VLD* reduces to 0.07, because both evidences are dependent due to the common cause

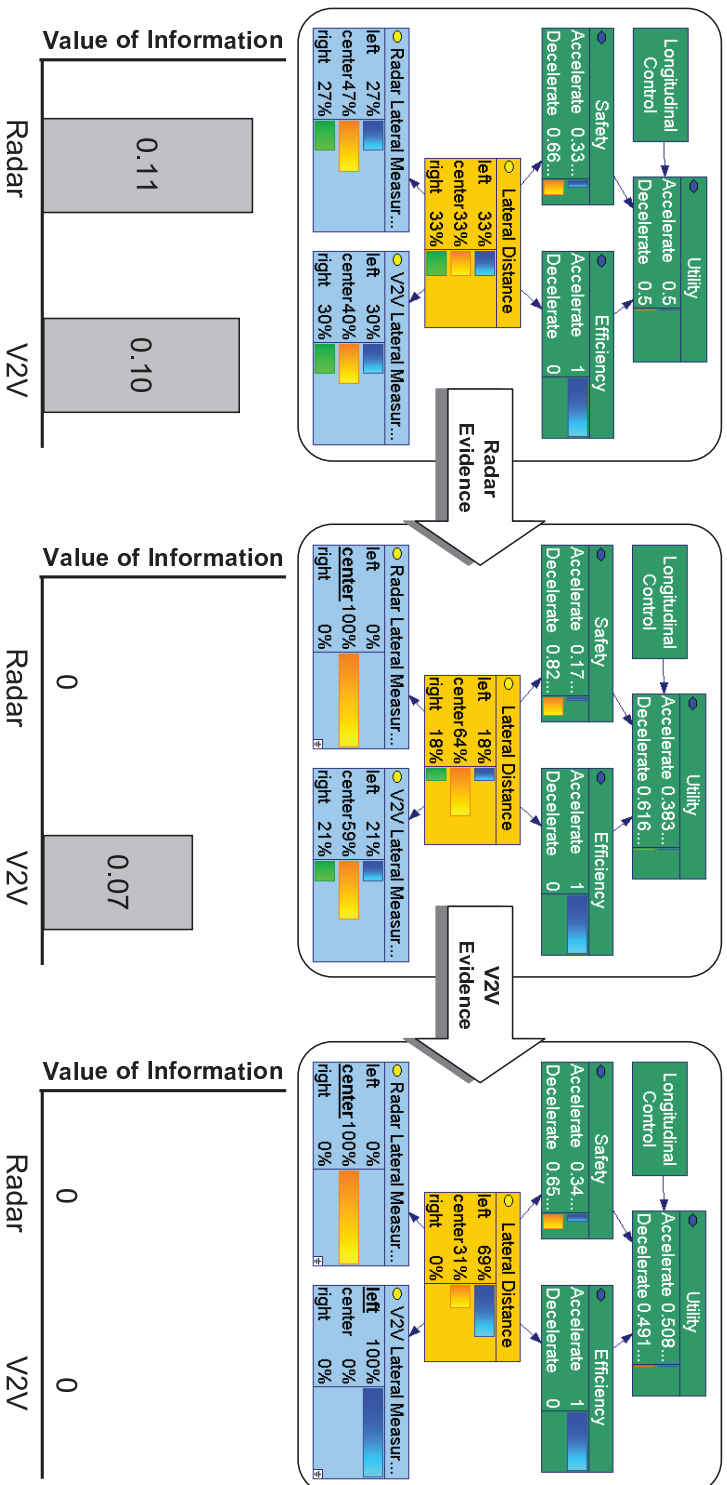


Figure 14: Variation of the value of information for different connection states of the Bayeslets for radar and V2V communications. Taken from [13].

*LD*. Thus, the belief in the *centre* state as the actual outcome of the *VLD* already has a probability of 59 %, see Figure 14 (middle).

The *VoI* of *VLD* is still positive however and, hence, is expected to add value for decision-making. This is justified, as the *MEU* can still change significantly, depending on the actual value of the second Bayeslet. For example, after the incorporation of the state *centre* for *RLD* a connection of the second Bayeslet to *VLD* may provide state *left*, as depicted in Figure 14 (right). This makes the *MEU* change to 0.51 for the *accelerate* action. Thus, the best action changes from *decelerate* to *accelerate* by the connection of the second Bayeslet.

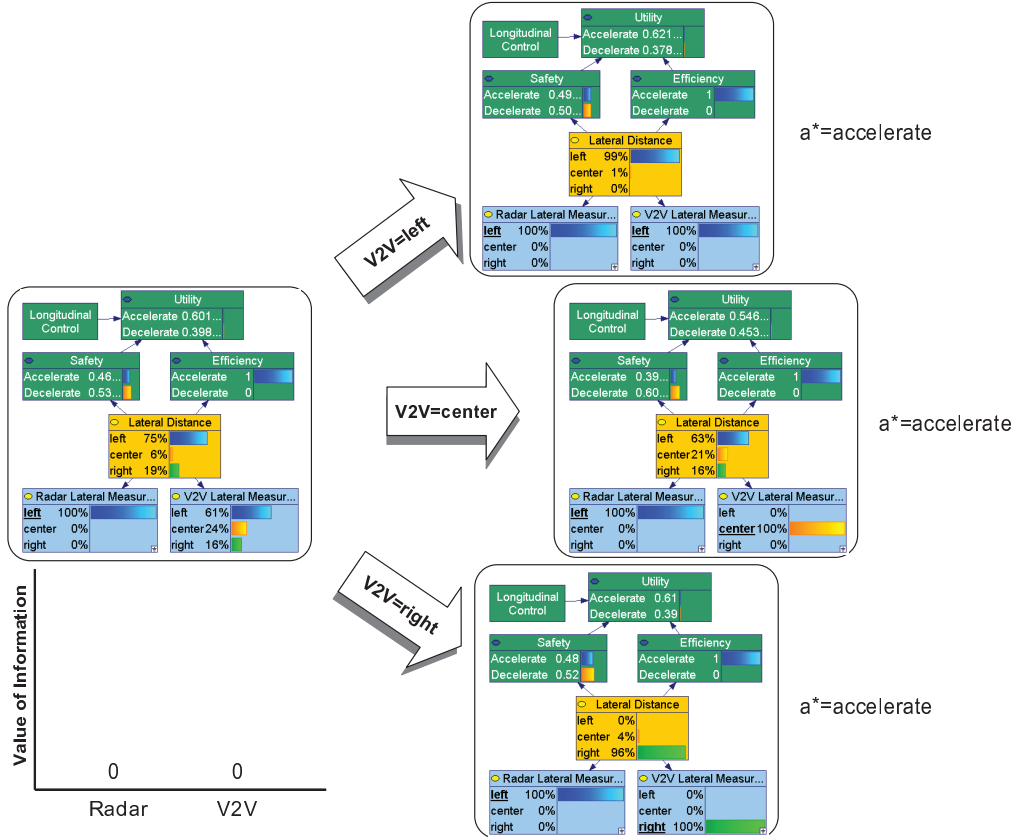


Figure 15: Variation of the maximum expected utility (MEU) with different outcomes of the V2V evidence after observing the radar evidence *left*. Taken from [13].

The *VoI* is not static for Bayeslets, but depends on the evidence already available. This can be seen in Figure 15. The figure is based on the same parameters and assumptions as Figure 14, but instead of *centre*, the outcome of the Bayeslet for *RLD* is *left*. Accordingly, the best action is *accelerate* with a *MEU* of 0.60. In this case  $VoI(VLD) = 0$ .

The figure shows on the right side the consequences of a hypothetical connection

of the Bayeslet for *VLD*. If its evaluation yields *left*, the best action obviously stays *accelerate*. If it provides *centre*, still *accelerate* is the best action and even if it provides *right*, the best action is *accelerate*. Thus, independently of the outcome of the Bayeslet the decision remains unchanged. Hence, it does not add information and this Bayeslet is not connected. In the case of the shown application example all transmissions of Bayeslet results could hence be neglected and also local evaluation is reduced to 50 %.

#### 5.4. Summary of Evaluation

This section has shown a simplistic example to illustrate the concepts presented in sections 3 and 4. We have seen that in this example we can save up to 50 % of inference time and 90 % of the related network traffic and, more importantly, that the complexity of inference is reduced significantly. By limiting the number of nodes included in an inference process, the factor responsible for the exponential growth in processing time, we approach a lower complexity class in computability theory. This makes initially more complex BNs computable in the first place.

## 6. Conclusion

This work has proposed the Bayeslet concept for structuring context inference networks which allows one to cope with very large amounts of information both from a computational and a remote communications point of view. Bayeslets represent enclosed and encapsulated knowledge domains that are, due to their size, easily manageable and processable and allow for distributed evaluation and selection of relevant information. The restrictions, reduced modelling flexibility and inference precision, have to be accepted in order to take advantage of tractable and bandwidth efficient context inference.

The approach presented for the composition of Bayeslets is novel and formal. It concretises the rather vague concept of *relevancy* used in many definitions of context. For a given objective, the most relevant information is selected and thereby, depending on an application specific threshold, information transmission is limited. This reduces inference time and increases bandwidth efficiency. In the presented simplistic example, we were able to show that in many cases the connection of a second Bayeslet is not useful. We could reduce transmissions over the network by about 90 % and reduce the inference time necessary to incorporate a Bayeslet to 50 %.

The determinant of the worth of information, utility, can be defined in various ways. Both approaches presented yield different results for the application example. This is due to the different objectives of both approaches. The objective of mutual information based on the entropy is always the reduction of uncertainty. The objective of the decision based approach, however, can be defined freely by the configuration of the utility node(s).

Consequently, both methods are not mutually exclusive, but are valid alternatives which can be chosen depending on the current situation. The creator/owner of a Bayeslet has the option to define decision and utility nodes for a specific objective. If no objective is defined, the standard would be the uncertainty reduction. This can be realised with the mutual information based composition decision.

In the future these results already confirmed in theory, have to be tested in real inter-vehicle communication. To this end, the Bayeslet management system has to be incorporated in the vehicle communication system.



A further step should be the consideration of maximum response times. They could be realised e.g. with an adaptive cost function for the composition of Bayeslets. In general, the work on cost functions for composition still offers room for research. A range of requirements (network response time, speed of computation, adaptation to situation, comparability among different Bayeslets, etc.) can be defined which has to be fulfilled for all different domains causing composition costs, such as time, memory and communication costs.

## Acknowledgements

This research has received funding from the European Communitys Seventh Framework Programme [FP7/2007-2013] under grant agreement no. 257493 of the Societies (Self Orchestrating Community ambiEnT IntelligEnce Spaces) Collaborative Project.

## References

- [1] B. van Arem, C.J. van Driel, R. Visser, The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics, *IEEE Transactions on Intelligent Transportation Systems* 7 (2006). IEEE Computer Society Press, Los Alamitos, CA, USA, ISSN: 1524-9050.
- [2] K.P. Baclawski, Bayesian network development, in: *International Workshop on Software Methodologies, Tools and Techniques, Keynote address.*, pp. 18–48.
- [3] O. Bangsø, M.J. Flores, F.V. Jensen, Plug & play object oriented bayesian networks, in: *Proceedings of the Tenth Conference of the Spanish Association for Artificial Intelligence (CAEPIA)*, volume 3040, Springer Verlag, 2003, pp. 457–467.
- [4] O. Bangsø, P.H. Wuillemin, Object Oriented Bayesian Networks: A Framework for Topdown Specification of Large Bayesian Networks and Repetitive Structures, Technical Report, Department of Computer Science, Aalborg University, Aalborg, Denmark, 2000.
- [5] B. Beamon, Evaluation of first order bayesian networks for context modeling & reasoning, in: *Proceedings of Third Annual Google Ph.D. Forum on Pervasive Computing and Communications*.
- [6] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, *the Journal of machine Learning research* 3 (2003) 993–1022.
- [7] E.J. Clarke, B.A. Barton, Entropy and MDL discretization of continuous variables for Bayesian belief networks, *International Journal of Intelligent Systems* 15 (2000) 61–92. John Wiley & Sons, Inc., New York, NY, USA, ISSN: 1098-111X.
- [8] G.F. Cooper, Probabilistic inference using belief networks is NP-hard, Technical Report KSL-87-27, Medical Computer Science Group, Knowledge Systems Laboratory, Stanford University, Stanford, CA, 1987.
- [9] G.F. Cooper, The computational complexity of probabilistic inference using bayesian belief networks, *Artif. Intell.* 42 (1990) 393–405.
- [10] M. Crotty, N. Taylor, H. Williams, K. Frank, I. Roussaki, M. Roddy, A pervasive environment based on personal self-improving smart spaces, Architectures and platforms for AMI, workshop on AmI08, European conference on Ambient Intelligence, in: *Architectures and platforms for AMI, workshop on AmI08, European conference on Ambient Intelligence*.
- [11] K. Frank, Adaptive and Tractable Bayesian Context Inference for Resource Constrained Devices. PhD Thesis, Waterford Institute of Technology, multicon multimedia consulting, Schöneiche, Germany, 2011. ISBN: 978-3930736188.
- [12] K. Frank, M. Röckl, T. Pfeifer, Optimizing dynamic composition of bayesian networks for context sensing and inference, in: *the 35th Annual IEEE Conference on Local Computer Networks (LCN 2010)*, Denver, Colorado, USA, pp. 312–315.
- [13] K. Frank, M. Röckl, T. Pfeifer, Intelligent information dissemination in collaborative, Context-Aware environments, in: *8th IEEE International Workshop on Managing Ubiquitous Communications and Services (MUCS 2011)*, Seattle, USA, pp. 287–293.
- [14] K. Frank, M. Röckl, P. Robertson, The Bayeslet concept for modular context inference, in: *Proceedings of Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies. UBICOMM08*, IEEE Computer Society Press, 2008, pp. 96–101.

- [15] T. Gu, H.K. Pung, D. Zhang, A Bayesian approach for dealing with uncertain contexts, in: Proceedings of the 2nd International Conference on Pervasive Computing (Pervasive '04), volume 176 of *Advances in Pervasive Computing*, Austrian Computer Society, 2004.
- [16] H. Guo, W. Hsu, A survey of algorithms for real-time Bayesian network inference, in: H. Guo, E. Horvitz, W.H. Hsu, E. Santos (Eds.), Working Notes of the Joint Workshop (WS-18) on Real-Time Decision Support and Diagnosis, AAAI/UAI/KDD-2002, AAAI Press, 2002.
- [17] I. Horrocks, P.F. Patel-Schneider, F.V. Harmelen, From SHIQ and RDF to OWL: The making of a web ontology language, *Journal of Web Semantics* 1 (2003) 2003.
- [18] C. Huang, A. Darwiche, Inference in belief networks: A procedural guide, *International Journal of Approximate Reasoning* 15 (1996) 225–263. Elsevier Science Publishers Ltd., ISSN: 0888-613X.
- [19] K.S. Hwang, S.B. Cho, Modular Bayesian networks for inferring landmarks on mobile daily life, in: AI 2006: Advances in Artificial Intelligence, 19th Australian Joint Conference on Artificial Intelligence, Hobart, Australia, December 4-8, 2006, Proceedings, volume 4304 of *Lecture Notes in Computer Science*, Springer Verlag, 2006, pp. 929–933.
- [20] K.S. Hwang, S.B. Cho, Modular Bayesian network learning for mobile life understanding, in: Proceedings of the 9th International Conference on Intelligent Data Engineering and Automated Learning, IDEAL '08, Springer Verlag, Berlin, Heidelberg, 2008, pp. 225–232.
- [21] F.V. Jensen, *Bayesian Networks and Decision Graphs*, Information Science and Statistics, Springer, 2001. ISBN: 0-387-95259-4.
- [22] D. Koller, A. Pfeffer, Object-oriented Bayesian networks, in: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI-97), Morgan Kaufmann Publishers Inc., 1997, pp. 302–313.
- [23] K.B. Laskey, S.M. Mahoney, Network fragments: Representing knowledge for constructing probabilistic models, in: D. Geiger, P.P. Shenoy (Eds.), UAI, Morgan Kaufmann, 1997, pp. 334–341.
- [24] D.V. Lindley, *Making decisions*, John Wiley & Sons, 2 edition, 1991.
- [25] D.J.C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003. ISBN: 0-521-64298-1.
- [26] D. Metzler, W.B. Croft, Linear feature-based models for information retrieval, *Information Retrieval* 10 (2007) 257–274.
- [27] M.A. Paskin, C.E. Guestrin, Robust probabilistic inference in distributed systems, in: Proceedings of the 20th conference on Uncertainty in artificial intelligence, UAI '04, AUAI Press, Arlington, Virginia, USA, 2004, pp. 436–445.
- [28] G. Pavlin, P.d. Oude, M. Maris, T. Hood, Distributed perception networks: An architecture for information fusion systems based on causal probabilistic models, in: Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, IEEE Computer Society Press, 2006, pp. 303–310.
- [29] J. Pearl, Fusion, propagation, and structuring in belief networks, *Artificial Intelligence* 29 (1986) 241–288. Elsevier Science Publishers Ltd., Essex, UK, ISSN: 0004-3702.
- [30] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Francisco, 1988. ISBN: 978-1558604797.
- [31] J. Pearl, *Causality: Models, Reasoning, and Inference*, Cambridge University Press, 2000. ISBN: 978-0521895606.
- [32] D.M. Pennock, Logarithmic time parallel Bayesian inference, in: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1998, pp. 431–438.
- [33] T. Pfeifer, Redundant positioning architecture, *Computer Communications* 28 (2005) 1575–1585. *Wireless Sensor Networks and Applications - Proceedings of the Dagstuhl Seminar 04122*. Elsevier Science Publishers B.V., North-Holland, ISSN: 0140-3664.
- [34] M. Pradhan, G.M. Provan, B. Middleton, M. Henrion, Knowledge engineering for large belief networks, in: R.L. de Mántaras, D. Poole (Eds.), UAI '94: Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence, July 29-31, 1994, Seattle, Washington, USA, Morgan Kaufmann, 1994, pp. 484–490.
- [35] S.E. Robertson, K. Sparck Jones, Relevance weighting of search terms, in: P. Willett (Ed.), *Document retrieval systems*, Taylor Graham Publishing, London, UK, 1988, pp. 143–160.
- [36] S.E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gatford, Okapi at trec-3, in: Proceedings of the Third Text REtrieval Conference (TREC 1994), pp. 109–126.
- [37] M. Röckl, *Cooperative Situation Awareness in Transportation*, Ph.D. thesis, University of Innsbruck (Austria), 2010.
- [38] M. Röckl, K. Frank, T. Strang, M. Kranz, J. Gacnik, J. Schomerus, Hybrid fusion approach combining autonomous and cooperative detection and ranging methods for situation-aware driver as-

- sistance systems, 2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, in: Proceedings of International Symposium on PIMRC 2008, IEEE, 2008.
- [39] M. Röckl, J. Gacnik, J. Schomerus, Integration of Car-2-Car Communication as a Virtual Sensor in Automotive Sensor Fusion for Advanced Driver Assistance Systems, in: A. of German Engineers (VDI) (Ed.), Proceedings of the FISITA 2008 World Automotive Congress, Springer Automotive Media, Munich, Germany, 2008, pp. 1–9.
- [40] M. Röckl, P. Robertson, Data dissemination in cooperative its from an information-centric perspective, in: IEEE International Conference on Communications (ICC), Cape Town, South Africa.
- [41] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995. ISBN: 978-0131038059.
- [42] L. Savage, The Foundations of Statistics, Courier Dover Publications, 2nd edition, 1972. ISBN: 978-0486623498.
- [43] C.E. Shannon, W. Warren, A Mathematical Theory of Communication, Bell system technical journal 27 (1948) 379–423. University of Illinois Press, ISSN: 0005-8580.
- [44] The University of Waitako, Weka 3: Data mining software in java, <http://www.cs.waikato.ac.nz/ml/weka/>, 1993-2008.
- [45] H. Tu, J. Allanach, S. Singh, K.R. Pattipati, P. Willett, Information integration via hierarchical and hybrid Bayesian networks, IEEE Transactions on Systems, Man and Cybernetics, Part A 36 (2006) 19–33. IEEE Computer Society Press, ISSN: 1083-4427.
- [46] Y. Xiang, D. Poole, M.P. Beddoes, Multiply sectioned Bayesian networks and junction forests for large knowledge-based systems, Computational Intelligence 9 (1993) 171–220. Blackwell Publishing Ltd., ISSN: 1467-8640.